ASCOS++: An Asymmetric Similarity Measure for Weighted Networks to Address the Problem of SimRank

HUNG-HSUAN CHEN, Computational Intelligence Technology Center,
Industrial Technology Research Institute, Taiwan
C. LEE GILES, Pennsylvania State University, University Park, US

In this article, we explore the relationships among digital objects in terms of their similarity based on vertex similarity measures. We argue that SimRank—a famous similarity measure—and its families, such as P-Rank and SimRank++, fail to capture similar node pairs in certain conditions, especially when two nodes can only reach each other through paths of odd lengths. We present new similarity measures ASCOS and ASCOS++ to address the problem. ASCOS outputs a more complete similarity score than SimRank and SimRank's families. ASCOS++ enriches ASCOS to include edge weight into the measure, giving all edges and network weights an opportunity to make their contribution. We show that both ASCOS++ and ASCOS can be reformulated and applied on a distributed environment for parallel contribution. Experimental results show that ASCOS++ reports a better score than SimRank and several famous similarity measures. Finally, we re-examine previous use cases of SimRank, and explain appropriate and inappropriate use cases. We suggest future SimRank users following the rules proposed here before naïvely applying it. We also discuss the relationship between ASCOS++ and PageRank.

Categories and Subject Descriptors: G.2.2 [**Mathematics of Computing**]: Discrete Mathematics—*Graph Theory*; G.1.3 [**Mathematics of Computing**]: Numerical Analysis—*Numerical linear algebra*; H.3.3 [**Information Systems**]: Information Storage and Retrieval—*Information search and retrieval*

General Terms: Theory, Algorithm

Additional Key Words and Phrases: Vertex similarity, SimRank, link prediction, link analysis, coauthor network, ASCOS++

ACM Reference Format:
Hung-Hsuan Chen and C. Lee Giles. 2015. ASCOS++: An asymmetric similarity measure for weighted networks to address the problem of simrank. ACM Trans. Knowl. Discov. Data 10, 2, Article 15 (October 2015), 26 pages.
DOI: http://dx.doi.org/10.1145/2776894

15

1. INTRODUCTION

Discovering similar objects in a social network and the link prediction problem remain active research areas [Aggarwal et al. 2012; Koutra et al. 2013]. Quantifying the similarity scores between every pair of nodes in a network can be the foundation for several research issues. For example, clustering similar nodes together may help recognize patterns of the network; the similarity scores among nodes may influence a network's evolution, growth, and decay.

This work is partially supported by the National Science Foundation and Dow Chemical.
Authors' addresses: H.-H. Chen, Computational Intelligence Technology Center, Industrial Technology Research Institute, Hsinchu; C. L. Giles, College of Information Sciences and Technology, Pennsylvania State University, University Park.

DOI: http://dx.doi.org/10.1145/2776894

To discover similar objects, researchers have proposed similarity measures based on the intrinsic properties of the objects, or the interaction/relationship among the objects. For the latter case, previous studies usually model the interaction/relationship among the objects by a network, and measure the similarity scores between objects by network topology based similarity measures. These measures can be classified into two categories: local structure based measures and global structure based measures. The local structure based measures quantify the similarity scores between nodes based on the local topology of the two target nodes. Thus, they are computationally efficient, but they utilize only limited information near the target nodes. On the other hand, the global structure based measures directly or indirectly ensemble the paths between nodes to gauge the similarity score. While they include more information, they are usually computationally costly in both time and space.

Among global structure based measures, SimRank [Jeh and Widom 2002] is probably the most popular and influential one. However, SimRank and its families or variations, such as SimRank++ [Antonellis et al. 2008], P-Rank [Zhao et al. 2009], and fast SimRank calculation [He et al. 2010; Li et al. 2010a, 2010b], suffer from a serious problem: if the length of a path between two nodes is an odd number, this path has no contribution to the final SimRank similarity scores [Chen and Giles 2013]. As a result, SimRank may output unreasonable scores under certain network topology. In extreme cases, the similarity score between neighbor nodes could be zero, even though they should be related to become neighbors. Readers who are interested in the guidelines of applying SimRank and possible methods to bypass the limitation of SimRank are encouraged to read Section 7 directly.

In this article, we thoroughly discuss the problem of SimRank, and address the problem by presenting a new similarity measure, ASCOS++, which enriches Asymmetric Network Structure Context Similarity (ASCOS) by including all paths between nodes and the weights of the edges along the paths in the calculation.

To compare the calculated similarity scores among different similarity measures, several studies assume that the tendency to form a link between nodes is a good proxy of similarity level. Thus, the performance of different similarity measures can be evaluated by the precision of the link prediction problem [Chen et al. 2011, 2012a; Liben-Nowell and Kleinberg 2007]: Given a network topology at time t, which measure will best predict future link formation? Researchers have applied several similarity measures on different networks to predict future behaviors or hidden relationship among nodes in the given networks. For example, future collaboration behaviors among scholars was predicted by analyzing coauthorship network [Chen et al. 2011; Liben-Nowell and Kleinberg 2007]; the hidden relationship among genetic diseases was inferred by studying the known relationships among genes and diseases [Chen et al. 2013a]; future co-starring behaviors among actors and actresses was indicated by their earlier co-starring behaviors [Chen et al. 2013b]. In one of our experiment, we compare different similarity measures on several networks by a similar setting. While future links in some networks are easier to predict and others are relatively difficult, ASCOS++ outperforms other measures in nearly all cases. We also conducted ASCOS++ on a word association network to show that ASCOS++ has an interesting property that is rarely seen in other similarity measures: ASCOS++ can identify the hierarchical relationship between nodes.

Since previous studies widely applied SimRank on different networks, we discussed several of these use cases carefully. We found that most previous studies blindly applied SimRank on inappropriate networks. As a result, they may reach a biased conclusion. If a user insists to use SimRank, we suggest following a procedure to modify the topology of the input network to circumvent the limitation of SimRank. Experimental results show that SimRank outputs better scores on the modified networks. However, the size

Table I. The SimRank Scores of the Toy Network Shown in Figure 1 (The Parameter c is Set to 0.9)

	N1	N2	N3
N1	1	0	0.9
N2	0	1	0
N3	0.9	0	1

of the modified network is usually much larger than the original network. Thus, it could be less efficient in computation.

The rest of the article is organized as follows. In Section 2, we review SimRank and several similarity measures. We also show the problem of SimRank and its families. Section 3 reviews our earlier proposed measure ASCOS. Section 4 introduces ASCOS++, which considers not only the topology of the network but also the edge weights into the measure. Section 5 proposes methods to efficiently compute ASCOS++. Section 6 compares ASCOS and ASCOS++ with several other popular similarity measures. Since SimRank is widely used in previous literature, we further discuss when SimRank could be utilized and when is inappropriate in Section 7. Summary and future works appear in Section 8.

2. RELATED WORKS

We introduce SimRank and several famous similarity measures in this section.

2.1. SimRank Introduction

The original SimRank defines the similarity score between two nodes in a recursive manner: *two objects are similar if they are referenced by similar objects* [Jeh and Widom 2002]. The base case is defined between a node and itself: a node is most similar to itself and the similarity value is defined to 1. Thus, SimRank score between nodes i and j ($i \neq j$) is defined by Equation (1).

$$s_{ij} := \frac{c}{|N(i)||N(j)|} \sum_{\forall k \in N(i)} \sum_{\forall \ell \in N(j)} s_{k\ell}, \tag{1}$$

where c is a user specified value to determine the relative importance between neighbors and neighbors of neighbors. The value of c is typically between 0 and 1, that is, neighbor nodes are usually more important than neighbors or neighbors. $N(i)$ returns the set of in-neighbors of node i, and $|\cdot|$ returns the size of the set.

SimRank influences several following methods. For example, P-Rank [Zhao et al. 2009] extends SimRank by considering both in-neighbors and out-neighbors. SimFusion [Xi et al. 2005] supports different intranode relations and different edge weights. SimRank++ [Antonellis et al. 2008] extends SimRank to weighted networks. The relationship between SimRank, P-Rank, and SimFusion is discussed in Cai et al. [2010].

2.1.1. The Problem of SimRank and its Families. Since the similarity score between two nodes i and j is dependent on the similarity scores between every other pairs of nodes, Equation (1) is calculated repeatedly until converges. In some cases, the SimRank scores converge to unreasonable values. For example, Table I lists the SimRank scores of the toy network shown in Figure 1. As can be seen, although node 1 is closer to node 2 but less closer to node 3, SimRank reports that the similarity score between node 1 and node 2 is smaller than node 1 and node 3.

Mathematically, the definition of SimRank is the same as measuring how soon two random surfers starting from i and j are expected to meet each other by randomly

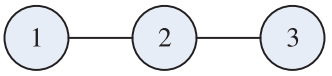

Fig. 1. A toy network in which SimRank converges to unreasonable values. For example, N1 can only reach N3 through N2, but the similarity score between N1 and N2 is smaller than the similarity sore between N1 and N3.

walking "backwards" in the graph. When setting the initial position of the two random surfers at node 1 and node 2 of the toy network, the two random surfers will never meet each other, since when one arrives at node 1 or node 3, the other must be at node 2. The same condition happens when we set the initial position of the two random surfers at node 2 and node 3. On the other hand, when setting the starting point to be at node 1 and node 3, they will meet each other after $2k + 1$ steps ($k = 0, 1, 2, \ldots$). Thus, SimRank returns counter-intuitive scores on this toy network: closer nodes are less similar than the nodes that are farther away.

In general, when two nodes can reach each other only through paths of odd lengths, the SimRank score between them is always zero. Even for a large connected network in which every node can almost surely reach every other node through several paths of even lengths, the other paths of odd lengths contribute nothing to SimRank calculation. Therefore, on average SimRank wastes half of the path information.

2.2. Other Topology Based Similarity Measures

This section reviews several popular vertex similarity measures.

2.2.1. Local Structure Based. Local structure based similarity measures utilize local network structures to decide the similarity score between two nodes. Similarity scores computed by this type of measures are usually proportional to the number of mutual neighbors between two target nodes and can be written as $s_{ij} = (|N(i) \cap N(j)|)/C$, where $N(i)$ is the set of neighbors of node i, $|X|$ returns the number of elements of set X, and C is a normalizing term whose value is determined by the specified similarity measure. For example, by Jaccard similarity the value of C is $|N(i) \cup N(j)|$ [Tan et al. 2006], by cosine similarity the value of C is $\sqrt{|N(i)||N(j)|}$ [Salton 1989], by topology overlapping the value of C is $\min(|N(i)|, |N(j)|)$ [Ravasz et al. 2002]. The Adamic–Adar measure [Adamic and Adar 2003] intentionally assigns more weights to the vertices with smaller degrees. Another local structure based similarity measure, Preferential Attachment [Barabási and Albert 1999], defines the similarity score between nodes by multiplying the degrees of two nodes. Empirical studies show that Preferential Attachment usually reports a poor performance. Comprehensive studies of local structure based measures can be found in Zhou et al. [2009] and Dong et al. [2011].

2.2.2. Global Structure Based. Global structure based measures consider the structure of the whole network to determine the similarity scores between pairs of nodes. For example, the Katz similarity is based on the total number of paths between two nodes in which longer paths are assigned lower weights [Katz 1953]. Compared to two low degree nodes, two nodes with very high degrees are more likely to have one or several paths of a fixed length ℓ between them. As a result, high degree nodes tend to be more similar to every other node by Katz measure. To address the problem, LHN [Leicht et al. 2006] suggested normalizing the number of paths of length ℓ by the expected number of such paths given the degrees of nodes. Recently, Chen et al. proposed RSS measure, which calculates similarity scores based on relation strength (a normalized edge weighting score) along all the paths between nodes [Chen et al. 2012a].

We found that the idea of the relaxed formulation in Zafarani et al. [2014] is very similar to our approach. However, there are still differences. The relaxed formulation was motivated to address the self-referential issue. ASCOS++, on the other hand, was motivated by the problem of SimRank: SimRank excludes the odd-paths during computation, which is undesirable. Another interesting difference is that the relaxed formulation, like most similarity measures, yields symmetric similarity scores, whereas ASCOS++ returns asymmetric similarity scores, because we column-normalize the adjacency matrix. In addition, the relaxed formulation needs to divide the adjacency matrix by the largest eigenvalue of the matrix to ensure convergence. Our proposed algorithm is more efficient because (1) we do not need to compute the eigenvalue of the adjacency matrix, and (2) we can solve the equations by the Jacobi iterative method, which possesses high degree of natural parallelism for distributed computation [Mehmood and Crowcroft 2005].

Global structure based similarity measure usually requires a great deal of computation. Several methods were proposed to approximate these measures. For the Katz score, a truncated spectral decomposition method was proposed [Acar et al. 2009]. For SimRank, an approximation measure [Li et al. 2010a] and a parallel computation [He et al. 2010] were investigated. Yu et al. proposed SimFusion+ [Yu et al. 2012] to prevent a divergence issue and faster computation. A comprehensive survey on both local structure based and global structure based similarity measures can be found in Lü and Zhou [2011].

2.2.3. Diffusion Based. A network diffusion captures how a subject *flows* from a node i to another node j. Researchers have studied network diffusion problems for decades and applied results on various domains, such as marketing, epidemic disease transmission, and rumor spreading [Chen et al. 2012; Kempe et al. 2003; Leskovec et al. 2007].

Intuitively, a node is more likely to transmit or receive a subject from closer nodes. Thus, the diffusion score can be a proxy of the vertex similarity score. Popular diffusion models include Independent Cascade (IC) model, in which a node has a single chance to transmit information to each of its neighbors, and Linear Threshold (LT) model, in which a node receives information if the summation of the influential power of its neighbors exceeds the threshold [Kempe et al. 2003]. Although these two diffusion models do not directly define the diffusion score, we may define such a score as the likelihood of successfully transmitting a message from a node i to a node j.

Kloster and Gleich suggested that a graph diffusion should follow Equation (2).

$$\mathbf{s} = \sum_{k=0}^{\infty} \alpha_k \mathbf{P}^k \mathbf{s}_0, \tag{2}$$

where α_k's are the decay factors, $\mathbf{s}_0$ is a column vector initialized by users, and $\mathbf{P} = [p_{ij}]$ is the row-normalized matrix of the adjacency matrix $\mathbf{A} = [a_{ij}]$ of the network (i.e., $p_{ij} = a_{ij} / \sum_{\forall \ell} a_{i\ell}$).

Based on the definition, the Katz score introduced in Section 2.2.2 can be regarded as a type of diffusion model, in which α_k is set to a fixed value c for all ks.

The heat kernel diffusion assigns the weights α_k as $t^k/k!$ [Chung 2007; Kloster and Gleich 2014]. Since the weight α_k decays very fast as k grows, two nodes that can only reach each other by a long path are unlikely to be similar to each other. As a result, several graph clustering algorithms applied heat kernel diffusion as the vertex similarity function for efficient computation [Chung 2009].

By Kloster and Gleich's definition of diffusion (Equation (2)), the value of α_k should be the same for every pair of nodes that are k steps away. Thus, ASCOS++ is close to, but not belongs to the diffusion-based similarity measure. As we will show in Section 4,

ASCOS++ may assign different α_k's to different pairs of nodes that are k steps away, depending on the edge weights of the path and the degrees of the nodes on the path.

2.2.4. Other Similarity Measures. Here we list few more similarity measures that do not belong to any of the previous categories. Based on archetypal analysis [Cutler and Breiman 1994], Tsourakakis defined each vertex as a convex combination of several extreme types of vertices, and measured vertex similarity via simplex fitting [Tsourakakis 2014]. Henderson et al. compared node similarity based on the nodes' role distributions, such as the tightly knit nodes, bridge nodes, and mainstream nodes [Henderson et al. 2012]. Thus, two nodes that are farther away may still be similar, if they serve similar structural roles. Agarwal and Chakrabarti proposed to combine Markovian random walk with relevance feedback information to rank nodes of a network [Agarwal and Chakrabarti 2007]. Some studies modeled the existence of a link as a supervised learning problem [Backstrom and Leskovec 2011; Chen et al. 2012b], in which the features can be topological features (e.g., the degree of a node) or intrinsic features (e.g., for a coauthorship network, the intrinsic feature could be an author's affiliation).

While it appears that most of the introduced similarity measures target at unweighted networks, several of these measures can deal with weighted networks with simple modification. For example, SimRank++ extends SimRank by modifying the underlying random walk model [Antonellis et al. 2008].

2.3. Graph Similarity

Instead of identifying similar nodes of a network, several works studied the *graph similarity*—the similarity between different graphs or networks. Researchers proposed utilizing simple graph static statistics to measure the similarity between two graphs. These static statistics could be, for example, the network size, the average degree, the average path length, the clustering coefficient, or the degree correlation of a network [Albert and Barabási 2002; Newman 2003]. Other studies tracked and compared the changes in networks over time [Caceres et al. 2011; Chen et al. 2013b; Leskovec et al. 2005]. Recently, many researchers showed tremendous interest in studying the topology of social networks, which typically contain millions to hundreds of millions of nodes. Thus, comparing two huge networks efficiently becomes a critical issue [Koutra et al. 2013].

3. ASCOS SIMILARITY MEASURE

The SimRank measure states that two nodes i and j are similar if the in-neighbors of i and the in-neighbors of j are themselves similar. The recursive definition employs the entire network topology in calculating the similarity scores. However, such a definition fails to capture the relationship between nodes that can only reach others in an odd number of steps, as pointed out in Section 1. In an extreme case, two neighboring nodes can have zero similarity score. This is counter-intuitive because neighboring nodes should have something in common if directly connected.

Instead of defining the similarity score between nodes by the relationship between both of the nodes' in-neighbors, ASCOS states that the similarity score from a node i to a node j is dependent on the similarity score from node i's in-neighbors to node j. This statement has two interesting properties. First, the calculation excludes the out-neighbors. Second, the similarity score is asymmetric because it considers the in-neighbors of i but not the in-neighbors of j. We justify these settings later.

We consider only the in-neighbors because an object is not defined by how it describes others but by how others describe it. This idea is very similar to PageRank since the incoming pages determines the importance of a page. However, we can easily extend this definition to consider both in-neighbors and out-neighbors.

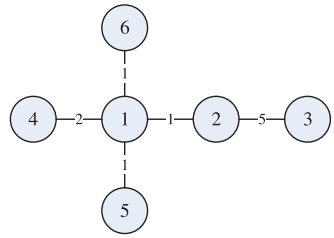

Fig. 2. The toy network.

Traditionally, a similarity measure defines the similarity function to be proportional to the inverse of the distance function, which is usually calculated by projecting the target objects into an n-dimensional coordinate system and measuring their distance. Thus, the similarity function should be symmetric, that is, $s_{ij} = s_{ji}$. By the definition, a statement of "a is like b" and a statement of "b is like a" should be of equal value. However, studies have shown that dimensional representations are not appropriate for some objects, like faces, countries, or personalities [Tversky 1977]. People tend to be more positive to "a is like b" than "b is like a" when b is more salient or general than a [Tversky 1977]. For example, "an ellipse is like a circle" is more likely to be true psychologically than "a circle is like an ellipse".

Another way to look at the similarity of nodes in a network is to measure the tendency to form a link between nodes. As suggested in Chen et al. [2011], when modeling the coauthoring behavior as a coauthorship network, a young researcher is usually more eager to establish connections with strong researchers than the other way around. The similarity score, which is a proxy to measure the tendency of link formation, is apparently asymmetric because a young researcher is more willing to establish a link to an experienced researcher than vice versa.

To make the asymmetry concept clearer, let's examine node 1 and node 4 of Figure 2. Node 4 has only one neighbor node 1, but node 1 has four neighbors: node 2, node 4, node 5, and node 6. In such a scenario people tend to be more positive to "node 4 is similar to node 1" than "node 1 is similar to node 4" because node 4's only neighbor is node 1 but node 1 has three other alternative options.

3.1. ASCOS Calculation

We define the similarity value from i to j to be the discounted cumulative similarity score from all i's neighbors to j. ASCOS score s_{ij} from i to j can be written as follows.

$$s_{ij} := \begin{cases} \frac{c}{|N(i)|} \sum_{\forall k \in N(i)} s_{kj} & \text{if } i \neq j \\ 1 & \text{if } i = j, \end{cases} \tag{3}$$

where $N(i)$ is the set of in-neighbors of node i.

The relative importance parameter c is between 0 and 1. It controls the relative importance between the direct neighbors and indirect neighbors, that is, neighbors' neighbors. The smaller the value, the less important the indirect neighbors are.

Algorithm 1 lists the pseudo code of ASCOS calculation based on the recursive definition.

4. ASCOS++ SIMILARITY MEASURE

In this section, we target improving ASCOS by considering not only the topology of the network but also the edge weights as well. We illustrate two scenarios to explain the intuition of the new design.

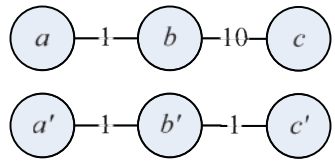

Fig. 3. A toy network with edge weights.

ALGORITHM 1: Naïve ASCOS calculation

Input: **A**: an adjacency matrix of size n by n; c: the discounted parameter
Output: $\mathbf{S} = [s_{ij}]$: ASCOS similarity score matrix

```
S ← initial guessing matrix of size n by n;
while S not converge do
    for i ← 1 to n do
        for j ← 1 to n do
            Update s_ij by Equation (3);
        end
    end
end
```

The first scenario is illustrated by Figure 3. In the top of the figure, the weight of the edge $e(a, b)$ is 1, and the weight of the edge $e(b, c)$ is 10. Apparently, the similarity score from b to c should be larger than the similarity score from b to a. Thus, the first mission is to include the absolute edge weight to the measure such that two nodes connected by an edge with a larger weight are more likely to be more similar than two nodes connected by an edge with a smaller weight.

The second scenario is slightly more complicated. Although the weight of the edge $e(a, b)$ and $e(a', b')$ are both 1, the fact that the weight of the edge $e(b, c)$ is 10 but the weight of edge $e(b', c')$ is 1 implies that the importance of a' for b' is larger than the importance of a for b, since node b has a relatively more important neighbor c. Thus, our second mission is to include the relative weight to the measure such that two nodes connected by an edge with a larger relative weight are more likely to be similar than two nodes connected by an edge with a smaller relative weight.

To incorporate the previous two missions into a similarity measure, we impose the *consistency rules* in the similarity scores. The definition of consistency rules slightly differs from the original definition in Antonellis et al. [2008], so that it can be applied on a general graph.

Definition 4.1 (*Consistency Rules*). Consider a weighted network $G = \{V, E\}$. Consider also four nodes $v_i, v_j, v_k, v_\ell \in V$ and two edges $e(v_i, v_j)$, $e(v_k, v_\ell) \in E$. We define w_{pq} as the weight of the edge $e(v_p, v_q)$. We further define $w_{p*} = \sum_{\forall r \in N(p)} w_{pr}$. We say the similarity scores s_{ij} and $s_{k\ell}$ follow consistency rules when the following statement is true:

If (1) $w_{ij} > w_{k\ell}$ and (2) $w_{ij}/w_{i*} > w_{k\ell}/w_{k*}$, then $s_{ij} > s_{k\ell}$.

The first condition and the second condition in the consistency rules address the two scenarios discussed earlier. We modified the ASCOS similarity score between two nodes i and j such that the similarity scores follow the consistency rules. The new similarity score, named ASCOS++, is defined by Equation (4).

$$s_{ij} := \begin{cases} c \cdot \sum_{\forall k \in N(i)} \frac{w_{ik}}{w_{i*}} (1 - \exp(-w_{ik})) s_{kj}, & \text{if } i \neq j \\ 1 & \text{if } i = j, \end{cases} \tag{4}$$

where w_{ik} is the weight of edge $e(i, k)$, and $w_{i*} = \sum_{\forall k \in N(i)} w_{ik}$.

ALGORITHM 2: Naïve ASCOS++ calculation

```
Input: A: an adjacency matrix of size n by n; c: the discounted parameter
Output: S = [s_ij]: ASCOS++ similarity score matrix
S ← initial guessing matrix of size n by n;
while S not converge do
    for i ← 1 to n do
        for j ← 1 to n do
            Update s_ij by Equation (4);
        end
    end
end
```

Equation (4) is defined for the following reasons. First, the term $1-\exp(-w_{ik})$ captures first condition in the consistency rule: when the value of w_{ik} is very large (e.g., $w_{ik} \to \infty$), the value of $1-\exp(-w_{ik})$ is close to 1, and when the value of w_{ik} is very close to zero, the value of $1-\exp(-w_{ik})$ is close to zero. Second, the term w_{ik}/w_{i*} captures the second condition in the consistency rule: when w_{ik} is close to w_{i*}, the value of $w_{ik}/wi*$ is close to 1, and when w_{ik} is much smaller than w_{i*}, the value of w_{ik}/w_{i*} is close to zero. Finally, since the values of c, $\sum_{\forall k \in N(i)}(w_{ik}/w_{i*}(1-\exp(-w_{ik})))$, and s_{kj} are all between 0 and 1, the product of them is still between 0 and 1.

Algorithm 2 shows the pseudo code of ASCOS++ calculation.

4.1. The Relationship between ASCOS++ and PageRank

PageRank is widely used to calculate the importance of a node in a network based on the network structure [Brin and Page 1998]. It is based on the intuition that a page p should be important if many important pages link to p. PageRank is formally defined by Equation (5).

$$PageRank(i) := \frac{1-c}{N} + c \sum_{\forall j \in N(i)} \frac{PageRank(j)}{|N(j)|}, \tag{5}$$

where c the damping factor, N is the number of nodes in the network, $N(i)$ is the set of in-neighbors of i.

The PageRank algorithm is commonly explained by the random surfer model: a surfer at a page follows one out-link page with probability c and jumps to a random page with probability $1-c$. The term $(1-c)/N$ explains the "random jumping" and ensures the recursive computation converges. However, if our target network is ergodic (i.e., all nodes are aperiodic and positive recurrent), we may ignore the term $(1-c)/N$ and still ensure convergence. Thus, a slightly simplified version of PageRank is shown in Equation (6).

$$PageRank(i) := c \sum_{\forall j \in N(i)} \frac{PageRank(j)}{|N(j)|}. \tag{6}$$

Equation (6) is very similar to Equation (3) when $i \neq j$. In fact, PageRank can be considered as an averaged ASCOS score. We explain the concept by the random surfer model: the ASCOS score from i to j is the same as measuring how soon a random surfer starting at i arrives at j, whereas the PageRank score of j is proportional to how soon a surfer starting at a *random* node i is expected to arrive at j. Thus, we can infer the

PageRank score of a node j by the ASCOS scores, as shown in Equation (7).

$$PageRank(j) \propto \frac{1}{N} \sum_{\forall i} s_{ij}, \tag{7}$$

where s_{ij} is the ASCOS similarity score from i to j.

5. EFFICIENT ASCOS++ COMPUTATION

Let n denotes the number of nodes in the given network, $\bar{N}$ represents the average number of in-neighbors of a node, and k be the required iterations to converge, Algorithm 2 takes $O(k\bar{N}n^2) \approx O(n^2)$ computation time (assuming $k \ll n$ and $\bar{N} \ll n$). This is infeasible when n the number of nodes is large. Motivated by Chen and Giles [2013], we re-write the equations to compute ASCOS++, and propose an efficient distributed computation approach.

5.1. Distributed ASCOS++ computation

Given a graph G and its adjacency matrix $\mathbf{A} = [a_{ij}]$, we compute $\mathbf{P} = [p_{ij}]$ as the column-normalized matrix of $\mathbf{A}$. that is, $p_{ij} = a_{ij} / \sum_{\forall k} a_{kj}$. When iterating Equation (3) to a sufficiently large number of times, the equation can be re-written in the form of Equation (8).

$$\mathbf{S} = c\mathbf{P}^T\mathbf{S} + (1-c)\mathbf{I} \Rightarrow (\mathbf{I} - c\mathbf{P}^T)\mathbf{S} = (1-c)\mathbf{I}, \tag{8}$$

where $\mathbf{P}^T$ is the transpose of $\mathbf{P}$.

Similarly, we may re-write Equations (4)–(9).

$$(\mathbf{I} - c\mathbf{Q}^T)\mathbf{S} = (1-c)\mathbf{I}, \tag{9}$$

where $\mathbf{Q} = [q_{ij}] = [p_{ij}(1 - \exp(-a_{ij}))]$.

By splitting $\mathbf{S}$ into n column vectors $\mathbf{S} = [\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_n]$ and the identity matrix $\mathbf{I}$ into n column vectors $\mathbf{I} = [\mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_n]$, we may re-write Equations (8) and (9) to Equations (10) and (11) respectively.

$$(\mathbf{I} - c\mathbf{P}^T)\mathbf{S}_i = (1-c)\mathbf{I}_i, i = 1, 2, \ldots, n \tag{10}$$

$$(\mathbf{I} - c\mathbf{Q}^T)\mathbf{S}_i = (1-c)\mathbf{I}_i, i = 1, 2, \ldots, n \tag{11}$$

This turns both ASCOS and ASCOS++ into a classic systems of linear algebra equations, in which $\mathbf{I} - c\mathbf{P}^T$ and $\mathbf{I} - c\mathbf{Q}^T$ are both coefficient matrices with dimension n by n; $\mathbf{S}_i$ is an unknown column vector with n variables to be solved; and $(1-c)\mathbf{I}_i$ is an constant column vector of size n.

We divide the original problem—solving the matrix $\mathbf{S}$ of dimension n by n—into n independent tasks: solving n column vectors $\mathbf{S}_i$ $(i = 1, 2, \ldots, n)$. Since both the matrix $(\mathbf{I} - c\mathbf{P}^T)$ in Equation (10) and the matrix $(\mathbf{I} - c\mathbf{Q}^T)$ in Equation (11) are diagonally dominant, (i.e., the magnitude of the diagonal entry in every row is larger than or equal to the sum of the magnitudes of all the nondiagonal entries in that row), we can determine the solutions by applying the Jacobi iterative method [Saad 2003]. As shown in Chen and Giles [2013], if we want to determine the similarity score between *one* pair of nodes (instead of *all* pairs of nodes), we only need to compute $1/n$ of the entire network. Therefore, the time complexity of obtaining the similarity score between one pair of nodes is $O(n)$.

Since the n tasks are independent, they possess high degree of natural parallelism. If n machines are available, we can obtain the similarity scores between *all* pairs of nodes in the network with time complexity $O(n)$.

6. EXPERIMENTS

Given a large network, it is difficult to evaluate the similarity scores returned by different similarity measures. To help users judge whether the returned scores are reasonable or not, we first conduct experiments on a toy network with small size. Next, we will compare different similarity measures based on their performance on link prediction problem. Finally, we showed that the asymmetric property implies the hierarchical relationship among nodes by applying ASCOS++ on a word association network.

6.1. The Equations of the Baseline Approaches

We list the equations of the baseline approaches in this section. We ignore SimRank equation and Jaccard equation since these equations were introduced in Section 2.

The RSS measure is calculated based on relation strength, a normalized edge weighting score defining the relative degree of similarity between neighboring vertices [Chen et al. 2012a]. The RSS measure is computed by Equation (12).

$$s_{ij} := \begin{cases} \prod_{k=1}^{K} r_{i,k+1} & \text{if } k \leq d \\ 0 & \text{otherwise,} \end{cases} \tag{12}$$

where $r_{k,k+1} = w_{k,k+1}/w_{k,*}$, d is the discovering range factor specified by users. For a weighted network, $w_{k,k+1}$ is the edge weight between node k and node $k+1$, and $w_{k,*}$ is the weighted degree of node k. For an unweighted network, $w_{k,k+1}$ is always 1 and $w_{k,*}$ is the degree of node k. Node k and $k+1$ are two neighboring nodes along any paths from node i to node j.

The Katz similarity is based on the total number of paths between two nodes where longer paths are assigned lower weights [Katz 1953]. Formally, Katz similarity is defined by Equation (13).

$$s_{ij} := \sum_{\ell=0}^{\infty} c^{\ell} \Phi(v_i, v_j, \ell), \tag{13}$$

where the function $\Phi(v_i, v_j, \ell)$ returns the number of paths of length ℓ between node v_i and node v_j.

In practice, the Katz similarity is usually calculated by Equation (14).

$$s_{ij} := \left[\left(\mathbf{I} - \frac{c}{\lambda_1} \mathbf{A} \right)^{-1} \right]_{ij}, \tag{14}$$

where $\mathbf{I}$ is the identity matrix, $\mathbf{A}$ is the adjacency matrix, and λ_1 is the largest eigenvalue of $\mathbf{A}$.

The LHN suggested normalizing the number of paths of length ℓ by the expected number of such paths given the degree of nodes, as expressed by Equation (15).

$$s_{ij} := \frac{2e\lambda_1}{m_i m_j} \left[\left(\mathbf{I} - \frac{c}{\lambda_1} \mathbf{A} \right)^{-1} \right]_{ij} \propto \frac{1}{m_i m_j} \left[\left(\mathbf{I} - \frac{c}{\lambda_1} \mathbf{A} \right)^{-1} \right]_{ij}, \tag{15}$$

where e is total number of edges, and m_i and m_j are the degrees of node i and node j respectively.

6.2. A Comparison of Different Similarity Measures on a Toy Network

Figure 4 shows a simple weighted network with only 6 nodes and 5 edges. Ideally, the similarity scores between every pair of nodes in a network should follow the *distance*

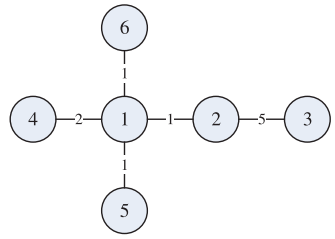

Fig. 4. A toy weighted network.

Table II. The ASCOS++ Scores of the Toy Network in Figure 4 (the Parameter c is Set to 0.9)

	N1	N2	N3	N4	N5	N6
N1	–	0.181	0.137	0.371	0.172	0.172
N2	0.284	–	0.760	0.105	0.049	0.049
N3	0.253	0.894	–	0.094	0.044	0.044
N4	0.778	0.141	0.107	–	0.134	0.134
N5	0.569	0.103	0.078	0.211	–	0.098
N6	0.569	0.103	0.078	0.211	0.098	–

Table III. The ASCOS Scores of the Toy Network in Figure 4 (the Parameter c is Set to 0.9)

	N1	N2	N3	N4	N5	N6
N1	–	0.573	0.347	0.530	0.530	0.530
N2	0.756	–	0.606	0.400	0.400	0.400
N3	0.681	0.9	–	0.360	0.360	0.360
N4	0.9	0.516	0.313	–	0.477	0.477
N5	0.9	0.516	0.313	0.477	–	0.477
N6	0.9	0.516	0.313	0.477	0.477	–

rule: the closer nodes tend to be more similar than the nodes that are farther. We apply several famous similarity measures on this toy network to examine whether their returned scores follow the distance rule.

Table II shows the result of ASCOS++. As can be seen, closer node pairs are more similar than the farther nodes. For example, the second row in the table indicates that node 2 is more similar to its neighbors (node 1 and node 3), and less similar to its neighbors' neighbors (node 4, node 5, and node 6). In addition, edge weights play an important role in the calculation. For example, similarity score from node 2 to node 3 is larger than node 2 to node 1, since the weight of the edge e(N2, N3) is larger than the weight of the edge e(N1, N2).

Table III shows the result of ASCOS. The result also follows the distance rule, since the similarity scores between closer nodes are larger than the scores between farther nodes. However, the edge weights are not included in the model. As can be seen, the similarity score from node 2 to node 1 is larger than the score from node 2 to node 3, even though edge e(N2, N3) has a higher weight.

Table IV, Table V, and Table VI show the scores of three other popular global structure based similarity measures: SimRank, Katz [1953], and LHN [Leicht et al. 2006]. The SimRank scores and LHN scores do not follow the distance rule, that is, in some cases the similarity score between closer nodes is smaller than the score of farther nodes. The Katz measure follows the distance rule in the example. However, the Katz scores are not normalized. It could be less convenient when the similarity scores need to be integrated with other scores.

Table IV. The SimRank Scores of the Toy Network in Figure 4 (the Parameter c is Set to 0.9). The Scores Do not Follow the Distance Rule, that is, the Closer Nodes Should be More Similar to the Nodes that are Farther Away. For Example, Although N1 can Only Reach N3 Through N2, $s_{1,3} > s_{1,2} = s_{2,3} = 0$, Which Violates the Distance Rule

	N1	N2	N3	N4	N5	N6
N1	–	0	0.759	0	0	0
N2	0	–	0	0.792	0.792	0.792
N3	0.759	0	–	0	0	0
N4	0	0.792	0	–	0.9	0.9
N5	0	0.792	0	.9	–	0.9
N6	0	0.792	0	.9	0.9	–

Table V. The Katz Scores of the Toy Network in Figure 4 (the Parameter c is Set to 0.9). It is Difficult to Normalize the Similarity Scores, Since the Maximum Score is Unknown in Advance

	N1	N2	N3	N4	N5	N6
N1	–	2.629	1.140	2.134	2.134	2.134
N2	2.629	–	1.144	1.140	1.140	1.140
N3	1.140	1.144	–	0.495	0.495	0.495
N4	2.134	1.140	0.495	–	0.926	0.926
N5	2.134	1.140	0.495	0.926	–	0.926
N6	2.134	1.140	0.495	0.926	0.926	–

Table VI. The LHN Scores of the Toy Network in Figure 4 (the Parameter c is Set to 0.9). The Scores Do not Follow the Distance Rule, that is, the Closer Nodes Should be More Similar to the Nodes that are Farther Away. For Example, Although N3 can Only Reach N4 Through N2 and N1, $s_{3,4} > s_{3,1}$, Which Violates the Distance Rule

	N1	N2	N3	N4	N5	N6
N1	–	0.329	0.285	0.533	0.533	0.533
N2	0.329	–	0.572	0.570	0.570	0.570
N3	0.285	0.572	–	0.495	0.495	0.495
N4	0.533	0.570	0.495	–	0.926	0.926
N5	0.533	0.570	0.495	0.926	–	0.926
N6	0.533	0.570	0.495	0.926	0.926	–

For example, researchers tend to look for potential collaborators within social circles and also share similar research interests [Chen et al. 2011]. We may leverage various similarity measures to quantify the social factor and obtain people within the researchers' social circles. However, we probably need other clues, such as the researchers' and potential collaborators' publication lists, to infer their research interests. To integrate the two factors—the social factor and the research interest overlapping, we probably want to normalize the scores of the two factors, so that the final scores will not be dominated by any of them.

Tables VII and VIII show the similarity scores calculated by RSS [Chen et al. 2012a] with and without including edge weights. Both of them follow the distance rule. However, sometimes the similarity score between neighbor nodes is 1, which could be confusing. In addition, even though the weight of the edge e(N1, N4) is 2 and the weight

Table VII. The RSS Scores of the Toy Network in Figure 4. It Could be Confusing When the Similarity Score Between a Node and its Neighbor Equals 1

	N1	N2	N3	N4	N5	N6
N1	–	0.25	0.125	0.25	0.25	0.25
N2	0.5	–	0.5	0.125	0.125	0.125
N3	0.5	1	–	0.125	0.125	0.125
N4	1	0.25	0.125	–	0.25	0.25
N5	1	0.25	0.125	0.25	–	0.25
N6	1	0.25	0.125	0.25	0.25	–

Table VIII. The Weighted RSS Scores of the Toy Network in Figure 4. It Could be Confusing When the Similarity Score Between a Node and its Neighbor Equals 1

	N1	N2	N3	N4	N5	N6
N1	–	0.2	0.167	0.4	0.2	0.2
N2	0.167	–	0.833	0.067	0.033	0.033
N3	0.167	1	–	0.067	0.033	0.033
N4	1	0.2	0.167	–	0.2	0.2
N5	1	0.2	0.167	0.4	–	0.2
N6	1	0.2	0.167	0.4	0.2	–

of the edge e(N1, N5) is 1, by weighted RSS the similarity score from Node 4 to Node 1 and from Node 5 to Node 1 are both 1.

6.3. A Comparison of Different Similarity Measures by Link Prediction

One popular way to evaluate different vertex similarity methods is by measuring their performance on link prediction problem: Given a snapshot of a network (called training network), which links will appear among these nodes in the future? Since new links are more likely to appear between similar nodes, a decent similarity measure should be able to predict future link formation to some extent. Previous literature sometimes evaluated different similarity measures by claiming the most similar n pairs of nodes will connect in the future. However, a reasonable estimation of the threshold value n is not always available. To capture the performance of overall prediction, instead of just the top-n prediction, two alternative measures are usually used: (1) the receiver operating characteristic (ROC) curve and (2) the precision-recall (PR) curve. The ROC curve shows the relationship between true positive rate (tpr) and false positive rate (fpr), and the PR curve, as the name suggested, shows the relationship between precision and recall. In this section, we use ROC curve to compare the performance of link prediction on various similarity measures, because theoretically ROC curves are not affected by the changes in the class distribution, and the judgment would be consistent even as imbalance becomes increasingly extreme [Lichtnwalter and Chawla 2012].

In this section, we employ two types of networks for evaluation. The first type includes six coauthorship networks from different academic areas. The second type is an item-relation network derived from a large-scale e-commerce website.

6.3.1. Coauthorship Network. We use the coauthorship network generated from the papers of the following six areas on arXiv[1] for link prediction. The six areas include Condensed Matter (cond-mat), High Energy Physics Experiment (hep-ex), High Energy

[1]http://arxiv.org/.

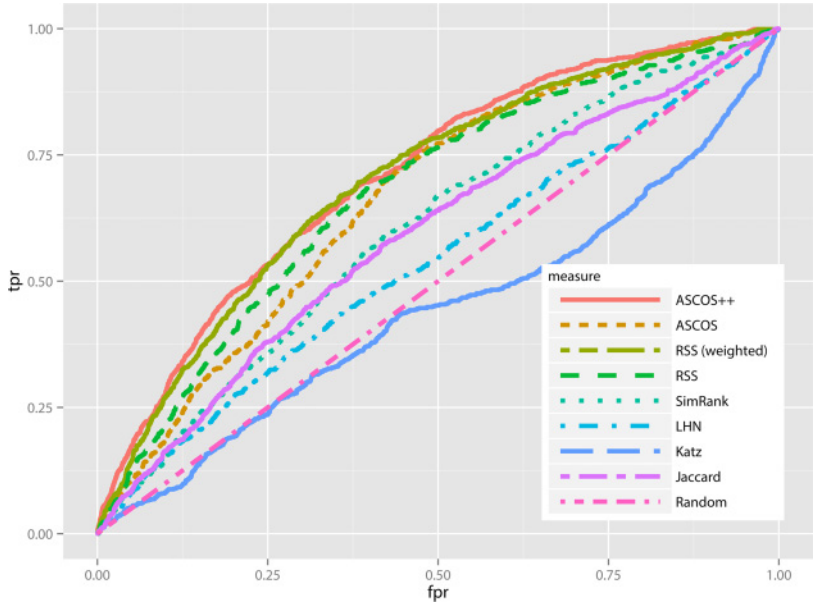

Fig. 5. The ROC curves of the link prediction results based on the cond-mat coauthorship network.

Physics Lattice (hep-lat), High Energy Physics Phenomenology (hep-ph), High Energy Physics Theory (hep-th), and Nuclear Experiment (nucl-ex) [Zhang et al. 2013]. Specifically, we use the authors who published between (1998, 2000) to generate the training network and predict the new coauthoring behavior in (2001, 2003). The weight of an edge $e(v_i, v_j)$ in the training network represents the number of coauthored papers between v_i and v_j during 1998 and 2000. The authors who published only in (2001, 2003) but not in (1998, 2000) are disregarded since their information is not available in the training network. In addition, since we are only interested in discovering the "new" links, the links that appear in both (1998, 2000) and (2001, 2003) are excluded from evaluation.

As shown in several studies, triadic closure is still a commonly observed property on networks, that is, the new links usually appear between nodes that were originally two steps away. This is probably also the reason that local structure based similarity measures seem to perform better than global structure based measures in several link prediction studies. Thus, in this section, we only report the similarity pairs that were originally two steps away. For example, assuming in the training graph node a and node b are 3 steps away, and node a and node c are two steps away. Even if by a certain vertex similarity measure $s(a, b) > s(a, c)$, we will set $s(a, b)$ to zero after the computation finishes. Note that even for the recursive based similarity measures, such as SimRank and ASCOS, setting $s(a, b)$ to zero will not affect the value of $s(a, c)$, because we set $s(a, b)$ to zero after the computation finishes, not during the computation process.

Based on the cond-mat coauthorship network, Figure 5 shows the ROC curves of several global structure based vertex similarity measures (SimRank, Katz, LHN, RSS with and without weight information, ASCOS, and ASCOS++) and one local structure based vertex similarity measure (Jaccard). The relative importance parameter c is set to 0.9 for ASCOS++, ASCOS, SimRank, Katz, and LHN. As seen, Katz performs poorly in predicting future collaborations, even though Katz outputs reasonable scores

Table IX. The AUCs of Different Measures Under Different Networks. Training Years: 1998–2000, Testing Years: 2001–2003

	cond-mat	hep-ex	hep-lat	hep-ph	hep-th	nucl-ex	Avg
ASCOS++ ($c = 0.9$)	**0.7089**	**0.9093**	0.6780	**0.8052**	**0.5993**	0.9292	**0.7717**
ASCOS ($c = 0.9$)	0.6644	0.8923	0.6452	0.7460	0.5332	0.9230	0.7340
RSS (weighted)	0.6985	0.8839	**0.6844**	0.7969	0.5691	0.9401	0.7622
RSS	0.6723	0.9013	0.6714	0.7824	0.5404	0.9435	0.7519
SimRank ($c = 0.9$)	0.6014	0.8674	0.5719	0.7032	0.4841	0.8847	0.6845
LHN ($c = 0.9$)	0.5396	0.8218	0.5654	0.6724	0.4774	0.9100	0.6644
Katz ($c = 0.9$)	0.4401	0.4161	0.5406	0.3000	0.5427	0.7299	0.4949
Jaccard	0.5895	0.8712	0.6569	0.7431	0.4888	**0.9441**	0.7156
Avg	0.6143	0.8204	0.6267	0.6937	0.5294	0.9006	

Table X. The AUCs of the Item Relation Network

	$p = 60\%$	$p = 70\%$	$p = 80\%$	$p = 90\%$	Avg
ASCOS++ ($c = 0.9$)	**0.6872**	**0.7908**	**0.8233**	0.8443	**0.7864**
ASCOS ($c = 0.9$)	0.6332	0.7685	0.8132	0.7740	0.7472
RSS (weighted)	0.6474	0.7247	0.8139	**0.8673**	0.7633
RSS	0.6232	0.7039	0.7963	0.8079	0.7328
SimRank ($c = 0.9$)	0.5757	0.6590	0.6898	0.7050	0.6574
LHN ($c = 0.9$)	0.5627	0.5835	0.5709	0.6786	0.5989
Katz ($c = 0.9$)	0.5473	0.5252	0.6149	0.6152	0.5757
Jaccard	0.5938	0.6448	0.7045	0.7475	0.6727
Avg	0.6088	0.6751	0.7284	0.7549	

in the toy network. ASCOS++ performs slightly better than RSS (with and without weight information), and much better than the rest methods. We show only the ROC curves for the cond-mat coauthorship network. However, we will show the AUC (Area Under Curve) of each ROC curve to quantify the comparison results. Note that when predicting future links based on asymmetric similarity measures (ASCOS++, ASCOS, RSS with and without edge weight information), we set the score between node a and node b to be the average of $s(a, b)$ and $s(b, a)$.

To quantify the prediction performance of each similarity measure, in Table IX we show the AUCs of these measures. The last row of Table IX lists the average AUC of different networks. As shown, the predictability for each network varies greatly: the new collaboration behaviors in nucl-ex field are highly predictable, whereas predicting future collaboration in the hep-th area is much challenging.

For each network, we highlight the measure that outputs the largest AUC value. As can be seen, even though the predictability of future collaboration differs from area to area, ASCOS++ outperforms other measures in nearly all cases.

6.3.2. Item Relation Network. We retrieved one-week web logs from an electronic commerce website, and built a network in which every node represents a product, and two nodes are connected by an edge with edge weight w if the two products were co-purchased by w customers during that week. We selected only the largest connected component of the network for experiment.

We evaluated the performance of different vertex similarity measures by their ability to predict missing links. We kept only $p\%$ of the edges to form the training network G_0. We applied different similarity measures on G_0 and claimed that the missing links should be at the non-neighboring node pairs with highest similarity scores.

Table X shows the AUCs of the ROC curves of different measures when p varies from 60 to 90. As can be seen, when more information is available, it appears that the unknown links are more likely to be correctly predicted. As for the link prediction

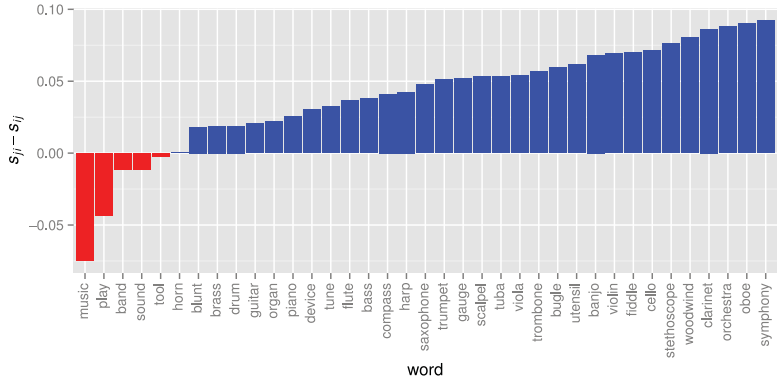

Fig. 6. The score differences of neighbor words of "instrument" to itself.

performance of different measures, the results are very consistent with the results on various coauthorship networks. In general, our proposed ASCOS++ better predicts unknown links in most cases.

6.4. Hierarchical Structure Inference

As discussed earlier, the ASCOS++ score s_{ij} from a node i to a node j tends to be smaller than s_{ji} if i is judged more salient or general than j. This asymmetric nature makes it possible to identify the hierarchical relationship between nodes in a network. To demonstrate this, we utilized the word association norms of over 10,000 words to generate a word relationship network, in which two words are connected if they are relevant based on a user survey [Nelson et al. 2004]. We evaluated the precision of selected terms and illustrate three cases to show the potential of inferring additional semantics between words without using any linguistics.

We selected 20 nouns as word i, and calculated the ASCOS++ value difference $s_{ji} - s_{ij}$ between word i and each of its neighbor word j. A positive value difference indicates that the word i is likely to be a super-class of the word j. We obtained the ground truth relationship (word i is super-class of word j, word i is sub-class of word j, or none of mentioned earlier) based on a user study. The user study shows that, based on ASCOS++, the precision of correctly identifying the hierarchical relation is 74.2%. The precision is defined by Equation (16).

$$\text{precision} := \frac{\text{\# of correctly predicted relations}}{\text{\# of test relations}}, \tag{16}$$

where a correctly predicted relation follow one of the two cases: (1) most users claim "word i is a super-class of word j" and $s_{ji} > s_{ij}$ by ASCOS++, or (2) most users claim "word i is a sub-class of word j" and $s_{ji} < s_{ij}$ by ASCOS++. If most users selected the option "none of above" for a pair of words, this pair of words is excluded in the evaluation.

We show three cases later for illustration. Figure 6 shows the first case: the ASCOS++ value difference $s_{ji} - s_{ij}$ given node i is the word "instrument" and node j is one of the 37 neighbor words of the node i. As shown, all the neighbor words representing musical instruments (such as trombone and cello) or other types of instruments (such as compass and stethoscope) have positive value differences. This implies that they are likely to be sub-classes of "instrument". Figure 7 demonstrates another example where node i is the word "fruit" and node j is one of its 62 neighbors. All fruits, such as cherry, peach, and kiwi, are successfully identified as the sub-classes of the word "fruit". The

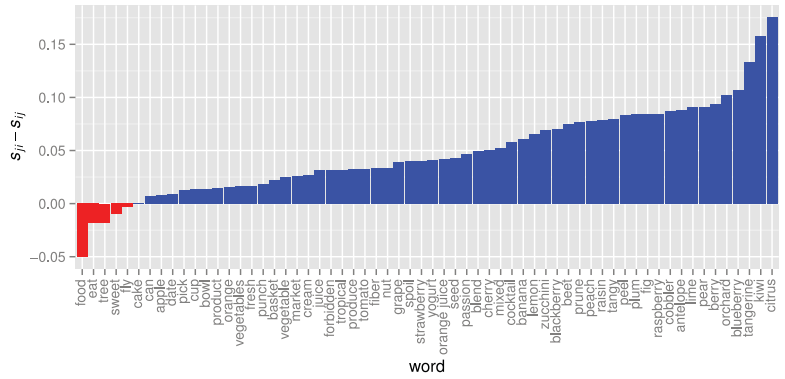

Fig. 7. The score differences of neighbor words of "fruit" to itself.

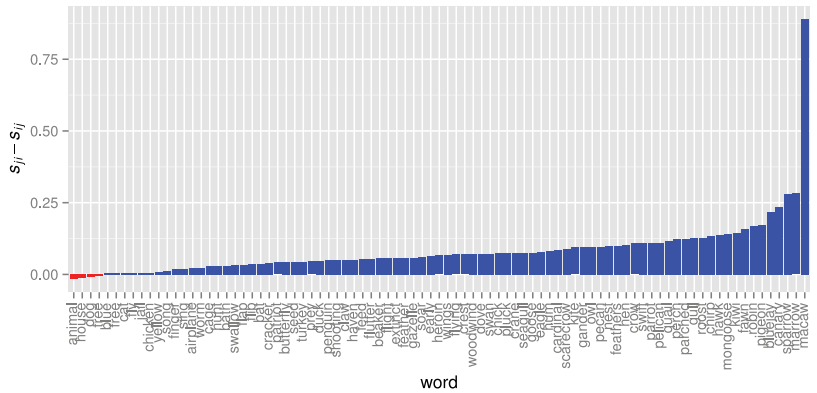

Fig. 8. The score differences of neighbor words of "bird" to itself.

third example, as shown by Figure 8, illustrates the relationship between the word "bird" to its neighbors. The results are similar to the previous two cases. However, there are a few words that are not sub-classes of node i, (for example, "symphony" and "orchestra" are not a kind of "instrument", and "cream" and "vegetable" are not one type of "fruit"). But in general, most results are reasonable. Since an ASCOS++ calculation involves no semantic information, the results could be further improved by semantic approaches.

6.5. Efficiency Comparison

6.5.1. Computing the Similarity Score Between One Pair of Nodes. To obtain networks of different sizes for experiment, we sampled nodes from the word association network by snowball sampling [Goodman 1961] until the number of sampled nodes reached the desired size. Specifically, we obtained 10 networks of sizes 1K, 2K, . . . , 10K.

For each sampled network, we logged the computation time of obtaining the similarity score between *one* pair of nodes based on SimRank, naïve ASCOS++ (Algorithm 2), and ASCOS++ based on Jacobi iterative technique (introduced in Section 5). We repeated each calculation 10 times and computed the average running time.

Figure 9 shows the result of relative computation time $r_x^{(n)} = t_x^{(n)}/t_x^{(1000)}$, where $t_x^{(n)}$ is the computation time of applying method x on a network of size n. Based on our analysis in Section 5, we expected that the running time of ASCOS++ based on Jacobi technique is linear to the network size, and the running time of SimRank and ASCOS++ (naïve)

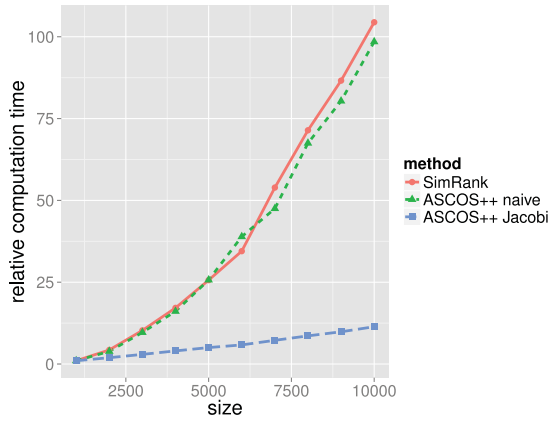

Fig. 9. The relative computation time of SimRank, ASCOS++ (naïve) and ASCOS++ (Jacobi technique) on 10 networks of size 1K, 2K, . . . , 10K.

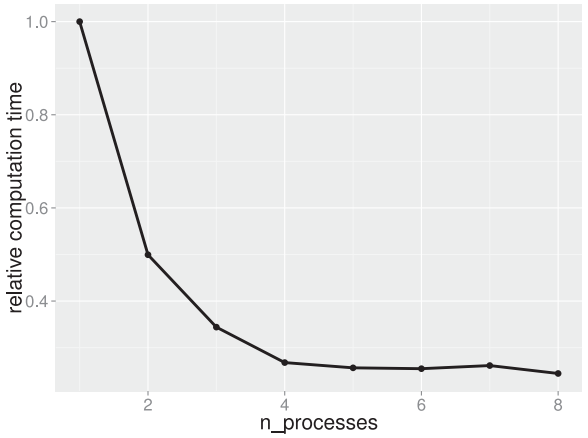

Fig. 10. The relative computation time of allocating $k (k = 1, 2, \ldots, 8)$ processes on a 4-core machine.

should be proportional to the square of the network size. The experimental results match the expectation.

6.5.2. Computing the Similarity Scores Between All Pairs of Nodes Under Distributed Environment. To validate the scalability of ASCOS++ based on Jacobi technique, we distributed the jobs of computing $\mathbf{S}_i$ $(i = 1, 2, \ldots, n)$ to k $(k = 1, 2, \ldots, 8)$ different processes on a 4-core machine and logged the running time.

Figure 10 shows the relative computation time $r^{(k)} = t^{(k)}/t^{(1)}$, where $t^{(k)}$ is the running time when allocation k processes. As can be seen, when k is larger than the number of cores 4, allocating more processes does not help much. However, when $k \leq 4$, the relative computation time is roughly $1/k$. This shows that ASCOS++ with Jacobi technique is highly scalable, as long as we have enough cores or enough machines.

7. MORE DISCUSSIONS ON SIMRANK

As we discussed earlier in this article, the main problem of SimRank is that it ignores the paths of even lengths between two nodes. As a result, SimRank may report counter-intuitive scores in certain types of networks. In this section, we propose possible

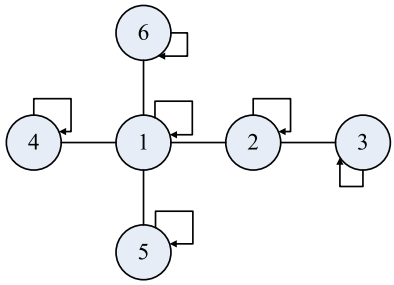

Fig. 11. The altered toy network with self-loops.

Table XI. The SimRank Scores of the Modified Toy Network with Self-Loops (The Relative Importance Factor c is Set to 0.9). Although Node 2 can only Reach Node 4 through Node 1, $s_{2,1} < s_{2,4}$, Which Still Violates the Distance Rule

	N1	N2	N3	N4	N5	N6
N1	–	0.557	0.528	0.622	0.622	0.622
N2	0.557	–	0.661	0.562	0.562	0.562
N3	0.528	0.661	–	0.478	0.478	0.478
N4	0.622	0.562	0.478	–	0.651	0.651
N5	0.622	0.562	0.478	0.651	–	0.651
N6	0.622	0.562	0.478	0.651	0.651	–

methods to mitigate the problem. After doing so, SimRank outputs a more reasonable score. However, previous works usually naïvely apply SimRank without further considering its limitation. As a result, their reported results might be questionable. If future studies need to apply SimRank on their network of interest, we suggest performing the operations suggested in this section before applying it.

7.1. Adding Self-Loops

When two nodes can only reach each other in an odd number of steps, by SimRank their similarity score is zero. One possible way to solve the problem is to alter the original network such that every node has a self-loop. As an example, the toy network shown in Figure 4 will become the one shown in Figure 11.

After adding a self-loop to every node, we can ensure that every node in the network can reach every other node in an even number of steps. Thus, the similarity scores between any pairs of the nodes in a connected graph will always be larger than zero. This seems to solve our earlier discussed problem. However, SimRank still suffers from the same problem fundamentally. Thus, only half of the paths between nodes contribute to the final SimRank score. As a result, SimRank may still output counter-intuitive scores. By applying SimRank on the modified graph (Figure 11), the similarity scores between all pairs of nodes are shown in Table XI. The new SimRank scores still do not follow the distance rule. For example, although node 2 must reach node 4 through node 1, the similarity score between node 2 and node 4 is larger than the similarity score between node 2 and node 1. As we said earlier, SimRank has shown to be the same as measuring how soon two random surfers starting from the two target nodes are expected to meet each other. Here we do not analyze the expected meeting time. Instead, we show only the probability that the two random surfers will meet each other in one-step. Readers who are interested in the expected meeting time may obtain these values by referring [Jeh and Widom 2002].

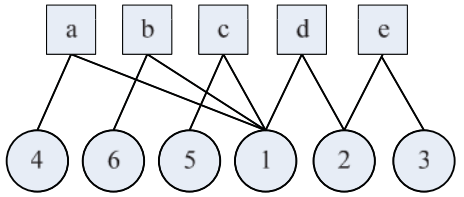

Fig. 12. The bipartite graph expanded from the toy network.

Table XII. The SimRank Scores of the Bipartite Network Expanded From the Toy Network (The Relative Importance Factor c is Set to 0.9)

	N1	N2	N3	N4	N5	N6
N1	–	0.497	0.378	0.646	0.646	0.646
N2	0.497	–	0.712	0.424	0.425	0.425
N3	0.378	0.712	–	0.330	0.330	0.330
N4	0.646	0.425	0.330	–	0.582	0.582
N5	0.646	0.425	0.330	0.582	–	0.582
N6	0.646	0.425	0.330	0.582	0.582	–

First, let the two random surfers starting at node 2 and node 4. Since node 2 has three neighbors (node 1, node 2, and node 3) and node 4 has two neighbors (node 1 and node 4), in the next step there are $3 \cdot 2 = 6$ possible positions of the two random surfers. Only when both of them reach node 1 they will meet each other. Thus, the probability that they will meet each other in one step is 1/6. When setting the initial positions of the two surfers be node 2 and node 1, there are $5 \cdot 3 = 15$ possible outcomes, and only both arrive at node 1 or both of them arrive at node 2 they will meet each other. Thus, the probability is 2/15. This shows that, even though node 1 and node 2 are neighbors, the probability that they will meet each other in the next step is smaller than setting the initial positions to node 4 and node 2.

7.2. Expanding to a Bipartite Graph

In last section, we modified the network such that there are paths of even lengths between every pair of nodes. We have showed that even though adding self-loop to every node can reach the goal, SimRank still outputs unreasonable scores because the paths of odd lengths, which might be valuable, are still excluded in the computation. In this section, we modify the network from another direction: modifying the original network such that the lengths of all paths between the nodes in the original are even numbers.

Specifically, for an edge $e(v_a, v_b)$ that connects node v_a and node v_b, we add a pseudo node v_{ab} and split edge $e(v_a, v_b)$ into two edges $e(v_a, v_{ab})$ and $e(v_{ab}, v_b)$. Thus, the toy network shown in Figure 4 becomes the one shown in Figure 12 (we ignore the edge weights here). All the circles in Figure 12 correspond to the nodes in Figure 4, and the edges in Figure 4 are split by the pseudo nodes represented by squares in Figure 12. As shown, every circle node can only reach other circle nodes in an even number of steps.

Table XII shows the SimRank similarity score between nodes after expanding the original graph into a bipartite graph. We show only node 1 to node 6 but not the pseudo nodes, since the pseudo nodes are out of our interest. As can be seen, the scores follow the distance rule: closer nodes have higher similarity scores. This is because every node of interest can only reach every other node of interest by paths of even lengths. Thus, all the paths between the nodes of interest are included in SimRank calculation.

Although the graph expanding technique discussed in the section is a work around solution to fix the problem of SimRank, the size of the expanded graph is much larger

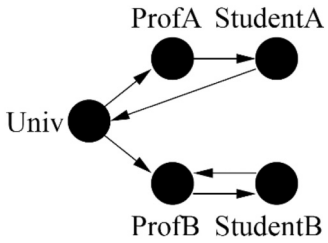

Fig. 13. The small web graph illustrated in Jeh and Widom [2002].

than the original graph. Specifically, if the original graph has n nodes and m edges, the expanded bipartite graph would have $n + m$ nodes and $2m$ edges. This will make the computation much less efficient.

7.3. Re-examination Use Cases of SimRank

While some previous studies applied SimRank on appropriate networks, most studies blindly utilized SimRank on any types of networks. Thus, they may obtain biased experimental results. In this section, we review several of these works and explain why or why not SimRank is appropriate in these networks. To close this section, we will introduce the general type of networks in which SimRank can be directly used.

In general, it is not appropriate to apply SimRank on web graph directly. In the original SimRank paper [Jeh and Widom 2002], the authors illustrated several examples to explain SimRank. Their first example, as shown in Figure 13, is a small web graph representing hyperlink relationship among five pages. The authors showed that by SimRank the seven most similar pairs of nodes in the network are: {ProfA, ProfB}, {StudentA, StudentB}, {Univ, ProfB}, {ProfA, StudentB}, {ProfB, StudentB}, {ProfB, StudentA}, and {Univ, StudentB}. However, SimRank fails to output the similarity relationship between the following three pairs of nodes: {Univ, ProfA}, {Univ, StudentA}, and {ProfA, StudentA}. One could check this by the random surfer model. For example, we could set the initial locations of the two random surfers at "Univ", and "ProfA". By randomly walking "backwards" in the graph, the two surfers can never meet each other. This violates our intuition: even though "Univ" directly links to "ProfA" and "ProfA" links to "Univ" through "StudentA", by SimRank the similarity score between "Univ" and "ProfA" is zero.

Several previous articles utilized SimRank to discover similar pairs or predict future link formation [Al Hasan and Zaki 2011; Liben-Nowell and Kleinberg 2007; Zhao et al. 2009; Zhou et al. 2009]. Their targeted networks are typically undirected, and links may appear between any pairs of nodes. As we said earlier, SimRank on average ignores half of the paths between nodes. Thus, applying SimRank directly on a general network is generally inappropriate.

Users' online shopping behaviors can usually be modeled by a bipartite graph: the first set of nodes is buyers and the second set of nodes is products. A buyer node connects to a product node if the buyer has purchased the product. In this setting, SimRank is appropriate to identify similar products. This is because every product node can only reach every other product node in even steps. Thus, all the paths between product nodes are included in SimRank calculation. Similarly, we could also utilize the bipartite graph to discover similar buyers. However, identifying similar {buyer, product} pairs in a bipartite graph is inappropriate, because every buyer node can only reach every other product node in odd steps. In general, if we can model the objects into a bipartite network with two sets such that links only exist between nodes of two different sets, it is appropriate to apply SimRank to measure the similarity scores between nodes within the same set. Jeh and Widom [2002] applied SimRank on shopping graph to infer

similar product pairs. Bao et al. [2007] utilized a variation of SimRank on "web page-annotation" bipartite graph to identify similar annotations or similar pages. These are the cases where SimRank can be used directly.

In general, if we could model the objects into a k-partite graph with k disjoint sets such that a node in the ith set only connects to nodes in the $(i-1)$th set or nodes in the $(i+1)$th set $(i = 2, 3, \ldots, k-1)$, it is appropriate to apply SimRank on the network to measure the similarity scores between nodes within the same set. This is because a node can only reach other nodes of the same set by a path of even length. Note that this graph is not a general k-partite graph, because for a general k-partite graph, a node in set k can connect to nodes in any other sets. Thus, a node may reach other nodes in the same set by a path with odd length. Such paths are disregarded by SimRank. Thus, simply modeling the objects into a general k-partite graph does not work here.

8. CONCLUSIONS AND FUTURE WORKS

SimRank and its family of methods have been widely utilized to identify similar pairs of nodes in a network. However, many of these studies did not realize the limitation of SimRank. In this article, we reported this problem with detailed explanation and examples. We presented a new similarity measure ASCOS++ that addresses this issue. The ASCOS++ measure considers both network topology and the edge weights. As a result, the returned scores follow both the distance rules and the consistency rules. We compared ASCOS++ with several popular similarity measures. Experimental results showed that ASCOS++ returns more reasonable similarity scores than other measures. In addition, we found that in virtually all our tested networks, ASCOS++ better predicts the missing or future links. The asymmetric scores of ASOCS++ helps discover the hierarchical relationship among nodes in a network. We also suggested methods to get through the limitation of SimRank, and proposed guidelines for future SimRank users.

Future work will further test ASCOS++ on various networks. It would be interesting to explore the relationships between ASCOS++, PageRank, SimRank, and several of these recursive network measures.

ACKNOWLEDGMENTS

We gratefully acknowledge valuable comments from reviewers.

REFERENCES

Evrim Acar, Daniel M. Dunlavy, and Tamara G. Kolda. 2009. Link prediction on evolving data using matrix and tensor factorizations. In *IEEE International Conference on Data Mining Workshops*. IEEE, Washington DC, USA, 262–269.

Lada A. Adamic and Eytan Adar. 2003. Friends and neighbors on the web. *Social Networks* 25, 3, 211–230.

Alekh Agarwal and Soumen Chakrabarti. 2007. Learning random walks to rank nodes in graphs. In *Proceedings of the 24th International Conference on Machine learning*. ACM, New York, NY, USA, 9–16.

Charu Aggarwal, Yan Xie, and Philip S. Yu. 2012. On dynamic link inference in heterogeneous networks. In *Proceedings of the 12th SIAM International Conference on Data Mining*. SIAM, Anaheim, CA, USA, 415–426.

Mohammad Al Hasan and Mohammed J. Zaki. 2011. A survey of link prediction in social networks. *Social Network Data Analytics*, 243–275.

Réka Albert and Albert-László Barabási. 2002. Statistical mechanics of complex networks. *Reviews of Modern Physics* 74, 1, 47–97.

Ioannis Antonellis, Hector Garcia Molina, and Chi Chao Chang. 2008. Simrank++: Query rewriting through link analysis of the click graph. *Proceedings of the VLDB Endowment* 1, 1, 408–421.

Lars Backstrom and Jure Leskovec. 2011. Supervised random walks: Predicting and recommending links in social networks. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*. ACM, Hong Kong, China, 635–644.

Shenghua Bao, Guirong Xue, Xiaoyuan Wu, Yong Yu, Ben Fei, and Zhong Su. 2007. Optimizing web search using social annotations. In *Proceedings of the 16th International Conference on World Wide Web*. ACM, Banff, Alberta, Canada, 501–510.

Albert-László Barabási and Réka Albert. 1999. Emergence of scaling in random networks. *Science* 286, 5439, 509–512.

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems* 30, 1, 107–117.

Rajmonda Sulo Caceres, Tanya Berger-Wolf, and Robert Grossman. 2011. Temporal scale of processes in dynamic networks. In *IEEE International Conference on Data Mining Workshops*. IEEE, Vancouver, Canada, 925–932.

Yuanzhe Cai, Miao Zhang, Chris Ding, and Sharma Chakravarthy. 2010. Closed form solution of similarity algorithms. In *Proceedings of the 33rd International SIGIR Conference on Research and Development in Information Retrieval*. ACM, Geneva, Switzerland, 709–710.

Hung-Hsuan Chen, Yan-Bin Ciou, and Shou-De Lin. 2012. Information propagation game: A tool to acquire humanplaying data for multiplayer influence maximization on social networks. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, Beijing, China, 1524–1527.

Hung-Hsuan Chen and C. Lee Giles. 2013. ASCOS: An asymmetric network structure context similarity measure. In *Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on*. IEEE, Niagara Falls, Canada, 442–449.

Hung-Hsuan Chen, Liang Gou, Xiaolong Zhang, and C. Lee Giles. 2011. CollabSeer: A search engine for collaboration discovery. In *Proceeding of the 11th Annual International ACM/IEEE Joint Conference on Digital Libraries*. ACM, Ottawa, Canada, 231–240.

Hung-Hsuan Chen, Liang Gou, Xiaolong Zhang, and C. Lee Giles. 2012a. Discovering missing links in networks using vertex similarity measures. In *The 27th ACM Symposium on Applied Computing*. ACM, Riva del Garda (Trento), Italy, 138–143.

Hung-Hsuan Chen, Liang Gou, Xioalong Zhang, and C. Lee Giles. 2012b. Predicting recent links in FOAF networks. In *Social Computing, Behavioral-Cultural Modeling and Prediction*. Springer, College Park, MD, USA, 156–163.

Hung-Hsuan Chen, Liang Gou, Xiaolong Luke Zhang, and C. Lee Giles. 2013a. Towards the discovery of diseases related by genes using vertex similarity measures. In *Healthcare Informatics (ICHI), 2013 IEEE International Conference on*. IEEE, Philadelphia, PA, USA, 505–510.

Hung-Hsuan Chen, David J. Miller, and C. Lee Giles. 2013b. The predictive value of young and old links in a social network. In *Proceedings of the ACM SIGMOD Workshop on Databases and Social Networks*. ACM, New York, NY, USA, 43–48.

Fan Chung. 2007. The heat kernel as the pagerank of a graph. *Proc. Natl. Acad. Sci. USA* 104, 50, 19735–19740.

Fan Chung. 2009. A local graph partitioning algorithm using heat kernel pagerank. *Internet Mathematics* 6, 3, 315–330.

Adele Cutler and Leo Breiman. 1994. Archetypal analysis. *Technometrics* 36, 4, 338–347.

Yuxiao Dong, Qing Ke, Bai Wang, and Bin Wu. 2011. Link prediction based on local information. In *Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on*. IEEE, Kaohsiung, Taiwan, 382–386.

Leo A. Goodman. 1961. Snowball sampling. *The Annals of Mathematical Statistics* 32, 1, 148–170.

Guoming He, Haijun Feng, Cuiping Li, and Hong Chen. 2010. Parallel simrank computation on large graphs with iterative aggregation. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, Washington, DC, USA, 543–552.

Keith Henderson, Brian Gallagher, Tina Eliassi-Rad, Hanghang Tong, Sugato Basu, Leman Akoglu, Danai Koutra, Christos Faloutsos, and Lei Li. 2012. Rolx: Structural role extraction & mining in large graphs. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, Beijing, China, 1231–1239.

Glen Jeh and Jennifer Widom. 2002. SimRank: A measure of structural-context similarity. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, Edmonton, Alberta, Canada, 538–543.

Leo Katz. 1953. A new status index derived from sociometric analysis. *Psychometrika* 18, 1, 39–43.

David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, Washington, DC, USA, 137–146.

Kyle Kloster and David F. Gleich. 2014. Heat kernel based community detection. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'14)*. ACM, New York, NY, USA, 1386–1395.

Danai Koutra, Joshua T. Vogelstein, and Christos Faloutsos. 2013. DeltaCon: A principled massive-graph similarity function. In *Proceedings of the SIAM International Conference on Data Mining*. SIAM, Austin, Texas, 162–170.

Elizabeth A. Leicht, Petter Holme, and Mark E. J. Newman. 2006. Vertex similarity in networks. *Physical Review E* 73, 2, 26120.

Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. 2005. Graphs over time: Densification laws, shrinking diameters and possible explanations. In *Proceedings of the 8th ACM International Conference on Knowledge Discovery and Data Mining*. ACM, Chicago, IL, USA, 177–187.

Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. 2007. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, San Jose, CA, USA, 420–429.

Cuiping Li, Jiawei Han, Guoming He, Xin Jin, Yizhou Sun, Yintao Yu, and Tianyi Wu. 2010a. Fast computation of simrank for static and dynamic information networks. In *Proceedings of the 13th International Conference on Extending Database Technology*. ACM, Lausanne, Switzerland, 465–476.

Pei Li, Hongyan Liu, Jeffrey Xu Yu, Jun He, and Xiaoyong Du. 2010b. Fast single-pair simrank computation. In *Proceedings of the SIAM International Conference on Data Mining*. SIAM, Columbus, Ohio, USA, 571–582.

David Liben-Nowell and Jon Kleinberg. 2007. The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology* 58, 7, 1019–1031.

Ryan Lichtnwalter and Nitesh V. Chawla. 2012. Link prediction: Fair and effective evaluation. In *Advances in Social Networks Analysis and Mining (ASONAM), 2012 IEEE/ACM International Conference on*. IEEE, Istanbul, Turkey, 376–383.

Linyuan Lü and Tao Zhou. 2011. Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and its Applications* 390, 6, 1150–1170.

Rashid Mehmood and Jon Crowcroft. 2005. Parallel iterative solution method for large sparse linear equation systems. *Computer Laboratory: University of Cambridge*.

Douglas L. Nelson, Cathy L. McEvoy, and Thomas A. Schreiber. 2004. The university of south florida free association, rhyme, and word fragment norms. *Behavior Research Methods, Instruments, & Computers* 36, 3, 402–407.

Mark E. J. Newman. 2003. The structure and function of complex networks. *SIAM REVIEW* 45, 2, 167–256.

Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. 2006. *Introduction to Data Mining*. Pearson Addison Wesley.

Erzsébet Ravasz, Anna Lisa Somera, Dale A. Mongru, Zoltán N. Oltvai, and A.-L. Barabási. 2002. Hierarchical organization of modularity in metabolic networks. *Science* 297, 5586, 1551–1555.

Yousef Saad. 2003. *Iterative Methods for Sparse Linear Systems*. SIAM.

Gerard Salton. 1989. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley.

Charalampos E. Tsourakakis. 2014. Towards quantifying vertex similarity in networks. *Internet Mathematics* 10, 3–4, 263–286.

Amos Tversky. 1977. Features of similarity. *Psychological Review* 84, 4, 327–352.

Wensi Xi, Edward A. Fox, Weiguo Fan, Benyu Zhang, Zheng Chen, Jun Yan, and Dong Zhuang. 2005. SimFusion: Measuring similarity using unified relationship matrix. In *Proceedings of the 28th Annual International SIGIR Conference on Research and Development in Information Retrieval*. ACM, Salvador, Brazil, 130–137.

Weiren Yu, Xuemin Lin, Wenjie Zhang, Ying Zhang, and Jiajin Le. 2012. SimFusion+: Extending simfusion towards efficient estimation on large and dynamic networks. In *Proceedings of the 35th International SIGIR Conference on Research and Development in Information Retrieval*. ACM, Portland, OR, USA, 365–374.

Reza Zafarani, Mohammad Ali Abbasi, and Huan Liu. 2014. *Social Media Mining: An Introduction*. Cambridge University Press.

Jun Zhang, Chaokun Wang, Philip S. Yu, and Jianmin Wang. 2013. Learning latent friendship propagation networks with interest awareness for link prediction. In *Proceedings of the 36th International ACM*

SIGIR Conference on Research and Development in Information Retrieval (SIGIR'13). ACM, Dublin, Ireland, 63–72.

Peixiang Zhao, Jiawei Han, and Yizhou Sun. 2009. P-Rank: A comprehensive structural similarity measure over information networks. In *Proceeding of the 18th ACM Conference on Information and Knowledge Management*. ACM, Hong Kong, China, 553–562.

Tao Zhou, Linyuan Lü, and Yi-Cheng Zhang. 2009. Predicting missing links via local information. *The European Physical Journal B-Condensed Matter and Complex Systems* 71, 4, 623–630.

Received August 2014; revised January 2015; accepted May 2015